Model and Feature Space Discriminative Training

Long Qin

August 19, 2011

1 Auxiliary Function

In the EM training of HMM parameters, we usually maximize an "auxiliary function" in the M-step. This auxiliary function is a function, when its value increases, the likelihood of the data given HMMs is bound to increase too. We maximize the auxiliary function instead of the likelihood function is because the auxiliary function is usually easier to directly maximize.

There are two kinds of auxiliary functions - the strong-sense auxiliary function and the weak-sense auxiliary function.

1.1 Strong-sense auxiliary function

If a function $F(\lambda)$ is to be maximized, then function $g(\lambda, \lambda')$ is a strong-sense auxiliary function for $F(\lambda)$ around λ', iff

$$g(\lambda, \lambda') - g(\lambda', \lambda') \leq F(\lambda) - F(\lambda'), \tag{1}$$

where $g(\lambda, \lambda')$ is a smooth function of λ. The strong-sense auxiliary function is the one we used in the EM algorithm. If g increases, then F also increases; if g is at a local maximum, then F is also at a local maximum. Therefore, repeated maximization of the auxiliary function is guaranteed to reach a local maximum of $F(\lambda)$.

1.2 Weak-sense auxiliary function

A weak-sense auxiliary function for $F(\lambda)$ is a smooth function $g(\lambda, \lambda')$ such that

$$\frac{\partial}{\partial \lambda} g(\lambda, \lambda')|_{\lambda=\lambda'} = \frac{\partial}{\partial \lambda} F(\lambda)|_{\lambda=\lambda'}. \tag{2}$$

So basically, the gradients of the two functions are the same around λ'. Although maximizing the auxiliary function $g(\lambda, \lambda')$ w.r.t. λ does not guarantee an increase of $F(\lambda)$, if the update converges (no change of λ), this implies that we have reached a local maximum of $F(\lambda)$ (the gradient is zero). Weak-sense auxiliary functions are useful when optimizing functions containing some terms that can be optimized by strong-sense auxiliary functions but others that cannot.

1.3 Smoothing function

A smoothing function around λ' is a smooth function of λ, $g(\lambda, \lambda')$, such that

$$g(\lambda, \lambda') \leq g(\lambda', \lambda') \tag{3}$$

for all λ. This smoothing function has its maximum at the initial point λ' (gradient is zero, so if a smoothing function around λ' is added to an objective function, the resulting function is a strong-sense function for that objective function around λ'. A smoothing function could also be added to a weak-sense auxiliary function to improve convergence without affective the local gradient.

2 Maximum Likelihood (ML)

In the ML training of HMMs, we want to find HMM parameters, so as to maximize the likelihood $p(O|s, \lambda)$,

$$F_{ML}(\lambda) = \log \sum_x f_x(\lambda), \tag{4}$$

where $f_x(\lambda)$ is the likelihood of a specific state sequences $p(O|s, \lambda, x)$. The strong-sense auxiliary function for $F(\lambda)$ is

$$g(\lambda, \lambda') = \sum_x \frac{f_x(\lambda')}{\sum_y f_y(\lambda')} \log(f_x(\lambda)). \tag{5}$$

As the first term in the summation is just the posterior probability of state sequence x, the auxiliary function could be re-written as

$$\begin{aligned} g(\lambda, \lambda') &= \sum_{j=1}^{J} \sum_{t=1}^{T} \gamma_j(t) \log N(o(t)|\mu_j, \sigma_j^2) && (6) \\ &= \sum_{j=1}^{J} -\frac{1}{2}\Big(\gamma_j \log(2\pi\sigma_j^2) + \frac{\theta_j(O^2) - 2\theta_j(O)\mu_j + \gamma_j\mu_j^2}{\sigma_j^2}\Big) && (7) \\ &= \sum_{j=1}^{J} Q\big(\gamma_j, \theta_j(O), \theta_j(O^2)|\mu_j, \sigma_j^2\big) && (8) \end{aligned}$$

The maximum occurs (set the gradient of Equ. 6 to 0), when $\mu_j = \frac{\theta_j(O)}{\gamma_j}$ and $\sigma_j^2 = \frac{\theta_j(O^2)}{\gamma_j} - \mu_j^2$.

3 Maximum Mutual Information (MMI)

The MMI training attempts to optimize the correctness of a model by formulating an objective function that penalizes the confusable models to the true model,

$$\begin{aligned} F_{MMI}(\lambda) &= \log \frac{p(O_r|s_r, \lambda)P(s_r)}{\sum_s p(O_r|s, \lambda)P(s)} && (9) \\ &= \log p(O|s^{num}, \lambda) - \log p(O|s^{den}, \lambda). && (10) \end{aligned}$$

As for the ML estimation, strong-sense auxiliary functions $g^{num}(\lambda, \lambda')$ and $g^{den}(\lambda, \lambda')$ could be derived separately. However, as the second term is negated, we can only apply weak-sense auxiliary function for the MMI objective function. To make the auxiliary function convex (it's concave when $\gamma_j^{num} < \gamma_j^{den}$), a smoothing function is added,

$$g^{sm}(\lambda, \lambda') = \sum_{j=1}^{J} Q\big(D_j, D_j\mu_j', D_j(\mu_j'^2 + \sigma_j'^2)|\mu_j\sigma_j^2\big). \tag{11}$$

The final auxiliary function becomes

$$\begin{aligned} g(\lambda, \lambda') &= g^{num}(\lambda, \lambda') - g^{den}(\lambda, \lambda') + g^{sm}(\lambda, \lambda') && (12)\\ &= \sum_{j=1}^{J} \Big(Q\big(\gamma_j^{num}, \theta_j^{num}(O), \theta_j^{num}(O^2)|\mu_j, \sigma_j^2\big) && (13)\\ &\quad - Q\big(\gamma_j^{den}, \theta_j^{den}(O), \theta_j^{den}(O^2)|\mu_j, \sigma_j^2\big) && (14)\\ &\quad + Q\big(D_j, D_j\mu_j', D_j(\mu_j'^2 + \sigma_j'^2)|\mu_j\sigma_j^2\big)\Big). && (15) \end{aligned}$$

Equ. 12 could be written as

$$\begin{aligned} g(\lambda, \lambda') &= \sum_{j=1}^{J} Q(t, X, S|\mu_j, \sigma_j^2) && (16)\\ &= \sum_{j=1}^{J} -\frac{1}{2}\Big(t\log(2\pi\sigma_j^2) + \frac{S - 2X\mu_j + t\mu_j^2}{\sigma_j^2}\Big), && (17) \end{aligned}$$

where

$$\begin{aligned} t &= \gamma_j^{num} - \gamma_j^{den} + D_j, && (18)\\ X &= \theta_j^{num}(O) - \theta_j^{den}(O) + D_j\mu_j', && (19)\\ S &= \theta_j^{num}(O^2) - \theta_j^{den}(O^2) + D_j(\mu_j'^2 + \sigma_j'^2). && (20) \end{aligned}$$

To maximize the MMI auxiliary function Equ. 12, we calculate

$$\frac{\partial g}{\partial \mu_j} = -\frac{1}{2} * (\frac{-2X + 2t\mu_j}{\sigma_J^2}) = 0 \tag{21}$$

and

$$\frac{\partial g}{\partial \sigma_j^2} = -\frac{1}{2} * (\frac{t}{\sigma_j^2} - \frac{S - 2X\mu_j + t\mu_j^2}{\sigma_j^4}) = 0. \tag{22}$$

So we can find

$$\mu_j = \frac{X}{t} = \frac{\theta_j^{num}(O) - \theta_j^{den}(O) + D_j\mu_j'}{\gamma_j^{num} - \gamma_j^{den} + D_j}, \tag{23}$$

then given $\mu_j = \frac{X}{t}$, we have

$$\begin{aligned} \sigma_j^2 &= \frac{S - 2x\mu_j + t\mu_j^2}{t} = \frac{S}{t} - 2\mu_j^2 + \mu_j^2 = \frac{S}{t} - \mu_j^2 && (24)\\ &= \frac{\theta_j^{num}(O^2) - \theta_j^{den}(O^2) + D_j(\mu_j'^2 + \sigma_j'^2)}{\gamma_j^{num} - \gamma_j^{den} + D_j} - \mu_j^2 && (25) \end{aligned}$$

4 fMMI

fMMI is a feature space discriminative training with the same objective function as the mode space MMI training. In fMMI, at first, a very high dimension feature vector h_t (Gaussian posteriors) is built; after applying a global matrix M, this high dimension feature vector is projected back the original feature space and added to the original features, such as

$$y_t = x_t + Mh_t. \tag{26}$$

Therefore, the fMMI objective function is

$$F_{MMI}(\lambda) = \log \frac{p(y_r|s_r,\lambda)P(s_r)}{\sum_s p(y_r|s,\lambda)P(s)}. \tag{27}$$

The feature transform M in Equ. 26 could be estimated by linear methods, such as gradient descent,

$$M_{ij} := M_{ij} + v_{ij}\frac{\partial F}{\partial M_{ij}}, \tag{28}$$

where v_{ij} is the parameter specific learning rate. Given Equ. 26, we can have

$$\frac{\partial F}{\partial Mij} = \sum_{t=1}^{T} \frac{\partial F}{\partial y_{ti}} h_{tj} \tag{29}$$

As we use gradient descent, instead of the extend Baum-Welch algorithm, to optimize the fMMI objective function $g(\lambda,\lambda')$, there is no smoothing function added,

$$g(\lambda,\lambda') = g^{num}(\lambda,\lambda') - g^{den}(\lambda,\lambda'). \tag{30}$$

Therefore, we have

$$\frac{\partial g}{\partial Mij} = \sum_{t=1}^{T} \frac{\partial g}{\partial y_{ti}} h_{tj} \tag{31}$$

Now, the question is how to calculate $\frac{\partial g}{\partial y_{ti}}$. This can be done similarly to the fMPE training,

$$\frac{\partial g}{\partial y_{ti}} = \frac{\partial g}{\partial y_{ti}}^{direct} + \frac{\partial g}{\partial y_{ti}}^{indirect}. \tag{32}$$

The first term in Equ. 32 can be calculated as

$$\frac{\partial g}{\partial y_{ti}}^{direct} = \frac{\partial g}{\partial \log p(y_{ti}|\lambda_i)} \frac{\partial \log p(y_{ti}|\lambda_i)}{\partial y_{ti}} \tag{33}$$

Considering the definition of auxiliary functions $g^{num}(\lambda,\lambda')$ and $g^{den}(\lambda,\lambda')$ in Equ. 6, we can easily get

$$\frac{\partial g}{\partial \log p(y_{ti}|\lambda_i)} = \gamma_i^{num}(t) - \gamma_i^{den}(t). \tag{34}$$

We can also easily compute

$$\frac{\partial \log p(y_{ti}|\lambda_i)}{\partial y_{ti}} = \frac{\mu_i - y_{ti}}{\sigma_i^2}. \tag{35}$$

Now, we have already computed the direct differential as

$$\frac{\partial g}{\partial y_{ti}}^{direct} = \big(\gamma_i^{num}(t) - \gamma_i^{den}(t)\big)\Big(\frac{\mu_i - y_{ti}}{\sigma_i^2}\Big). \tag{36}$$

The indirect differential is added here, because the features will affect the ML trained parameters. So during the optimization of M, we also take account the HMM parameters λ, so

$$\frac{\partial g}{\partial y_{ti}}^{indirect} = \frac{\partial g}{\partial \lambda_i}\frac{\partial \lambda_i}{\partial y_{ti}} = \frac{\partial g}{\partial \mu_i}\frac{\partial \mu_i}{\partial y_{ti}} + \frac{\partial g}{\partial \sigma_i^2}\frac{\partial \sigma_i^2}{\partial y_{ti}}. \tag{37}$$

As we can easily get

$$\frac{\partial \mu_i}{\partial y_{ti}} = \frac{\partial}{\partial y_{ti}}\frac{\sum_{t=1}^{T}\big(\gamma_i(t)y_{ti}\big)}{\gamma_i} = \frac{\gamma_i(t)}{\gamma_i} \tag{38}$$

and

$$\frac{\partial \sigma_i^2}{\partial y_{ti}} = \frac{\partial}{\partial y_{ti}}\big(\frac{\sum_{t=1}^{T}(\gamma_i(t)y_{ti}^2)}{\gamma_i} - \mu_i^2\big) = \frac{2\gamma_i(t)y_{ti}}{\gamma_i} - \frac{2\gamma_i(t)\mu_i}{\gamma_i} = \frac{2\gamma_i(t)(y_{ti}-\mu_i)}{\gamma_i}, \tag{39}$$

in addition, we already calculated $\frac{\partial g}{\partial \mu_i}$ and $\frac{\partial g}{\partial \sigma_i^2}$ in Section 3, the indirect differential is

$$\begin{aligned}
\frac{\partial g}{\partial y_{ti}}^{indirect} &= \frac{\gamma_i(t)}{\gamma_i}\Big(\frac{\partial g}{\partial \mu_i} + \frac{\partial g}{\partial \sigma_i^2}(y_{ti}-\mu_i)\Big) &(40)\\
&= \frac{\gamma_i(t)}{\gamma_i}\Big(\frac{X - d\mu_j}{\sigma_J^2} + \big(\frac{S - 2H\mu_j + d\mu_j^2}{\sigma_j^4} - \frac{d}{\sigma_j^2}\big)(y_{ti}-\mu_i)\Big) &(41)
\end{aligned}$$

where

$$\begin{aligned}
d &= \gamma_j^{num} - \gamma_j^{den}, &(42)\\
H &= \theta_j^{num}(x) - \theta_j^{den}(x), &(43)\\
S &= \theta_j^{num}(x^2) - \theta_j^{den}(x^2). &(44)
\end{aligned}$$

By far, we have already presented how to calculate the differential $\frac{\partial g}{\partial M_{ij}}$. The iterative training of the feature matrix is

- Starting from ML model λ_0, generate num + den lattices N, D
  - Iteration 1:
    - Phase 0: (needs λ_0, N, D, M_0 is 0)
      - Accumulate MMI and ML statistics
      - Compute $\frac{\partial g}{\partial \lambda_0}$ and ML Gaussian counts
    - Phase 1: (needs λ_0, N, D, $\frac{\partial g}{\partial \lambda_0}$)
      - Accumulate $\frac{\partial g}{\partial M_{ij}}$
      - Compute M_1 using gradient descent
    - Phase 2: (needs λ_0, N, M_1)
      - Accumulate ML statistics using new transform M_1
      - Compute new parameters λ_1
  - Iteration 2:
    - Etc.

5 Minimum Phone Error (MPE)

6 fMPE

7 Boosted-MMI

The objective function for boosted-MMI is

$$F_{BMMI} = \log \frac{p(O_r|s_r,\lambda)P(s_r)}{\sum_s p(O_r|s,\lambda)P(s)\exp(-bA(s,s_r))}, \tag{45}$$

where $A(s, s_r)$ is the raw phone accuracy of sentence s given the reference s_r, which equals the number of correct phones minus the number of insertions. As $A(s, s_r)$ is not a function of λ, the update formula of μ_j and σ_j^2 will be exactly the same as Equ. 23 and Equ. 24. The only difference between MMI and BMMI is that, when performing the forward-backward algorithm on the lattice, besides the acoustic score and language score, there is also a phone accuracy score $A(s, s_r)$ for each arc.

8 fBMMI

The fBMMI training is very similar to the fMMI training we discussed in Section 4. In fMMI, we try to optimize the feature transform with regard to the MMI objective function. However, in fBMMI, we optimize the feature transform with regard to the BMMI objective functions. Again, the training procedure and statistics accumulation in fBMMI in exactly the same as in fMMI. The only difference here is when performing forward-backward algorithm on the lattices, besides the acoustic score and language score, we also need to compute the raw phone accuracy for each arc.